

Linux Netatalk-HOWTO

Netatalk is a package that lets a Unix machine supply Appletalk print and file services on a LAN. The package supports AppleShare IP and classic Appletalk protocols. With netatalk, Macintosh computers can mount Unix volumes and print to Unix print spools as if they were standard Appletalk network devices.

This document is intended as a guide to help the Linux system administrator set up and administrate netatalk. Linux is a freely distributable POSIX compliant Unix for 386, 486, Pentium, PowerMacintosh, Sun and DEC Alpha hardware as well as several other experimental platforms. (SGI, etc.)

For those of you who are impatient, there is the Linux Netatalk-HOWTO: [Guide for the Impatient](#).

Netatalk is brought to you by our friends at The Research Systems Unix Group at The University of Michigan. [netatalk@umich.edu]

The original netatalk homepage is available at:

[<http://www.umich.edu/~rsug/netatalk/>]

Another place to look for netatalk info is Bill McGonigle's Faq-O-Matic at:

[<http://threepio.hitchcock.org/cgi-bin/faq/netatalk/faq.pl>]

AppleShare IP support (AFP/TCP) as well as the latest bug fixes and enhancements are included with Adrian Sun's version of netatalk called netatalk+asun. This HOWTO covers Adrian's version of netatalk as it is the most up to date. The "old version" of this HOWTO can be found at:

[<http://thehamptons.com/anders/netatalk/old/>]

To supply file and print services to Windows machines (using SMB over TCP) you should look into running Samba. Netatalk will happily co-exist with Samba. [<http://samba.isca.uiowa.edu/samba/samba.html>]

Netatalk+asun lets a Macintosh see a Linux box. To do the reverse (have the Linux box see the Mac) you need to install the **afpfs** module from:

[<http://www.odyssey.co.il/~heksterb/Software/afpfs/>] **Note:** The site doesn't seem to be working right now, but we have a mirror at:

- [<http://thehamptons.com/anders/netatalk/mirror/afpfs-1.0b1.tar.gz>]
- [<http://thehamptons.com/anders/netatalk/mirror/afpfs-1.0b2q2.tar.gz>]

Comments and corrections are very welcome. Someone had sent me a diff of my pages correcting my horrible spelling, and I proceeded to blow away my mailbox by mistake so I lost it. Anyone wanting to help out with that would be greatly appreciated.

If you have problems with netatalk, usually a good place to post them is to the linux-ataalk listserv. Subscribe to it by sending mail to listserv@netspace.org with no subject and a body which reads SUBSCRIBE LINUX-ATALK (Your Full Name). To unsubscribe, send mail to the same place with the message text UNSUBSCRIBE LINUX-ATALK. Posts to the listserv should be sent to linux-ataalk@netspace.org.

Before you begin:

- You might want to look at some hardware / software [suggestions](#).

What you will need:

- A computer running Linux with a 2.0.* or newer kernel (with source) and compilers.
- A copy of the latest netatalk+asun source from [<ftp://ftp.u.washington.edu/public/asun/>]
 - RedHat makes an RPM of netatalk+asun at [<ftp://contrib.redhat.com/libc6/i386/netatalk-1.4b2+asun2.1.0-5.i386.rpm>]
 - Debian makes binaries available at [<http://cgi.debian.org/www-master/debian.org/Packages/stable/net/netatalk.html>]
 - Original netatalk code (without TCP/IP support) can be found at [<ftp://terminator.rs.td.umich.edu/unix/netatalk/>]
- LAN with a Mac (don't really need this but then, what's the point?)

Decompress and untar the source code with the following commands:

```
elmer:~/src> gzip netatalk-1.4b2+asun2.0a18.2.tar.gz
elmer:~/src> tar -xvf netatalk-1.4b2+asun2.0a18.2.tar
or if you downloaded the rpm version:
```

```
eyore:~/src> rpm -i netatalk-1.4b2+asun2.1.0-5.i386.rpm
```

Edit the Makefile in the root of the source tree:

Set the destination directory:

DESTDIR=/usr/local/ataalk, which is the default is probably OK for most installations.

Comment out PAM support if you don't have it.

If you don't have PAM, (Pluggable Authentication Modules) you should comment out the line that reads PAMDIR=/usr. Basically RedHat uses PAM and Slackware doesn't. PAM makes it so you can "plug" user authentication modules (ways of checking usernames and passwords) into your system. Typical authentication modules would include shadow password support, NIS support, etc.

Edit the etc/afpd/Makefile if you need shadow password support. (PAM takes care of shadow password checking, so if you have PAM and a shadow password system, you can skip this step.)

Most major Linux distributions such as RedHat use shadow password suites. If you don't know if you have shadow passwords, look at your password file. If you have an "x" where the encrypted password should be, you have shadow passwords and you need to define shadow passwords in your Makefile. (Sample line from a shadow password file: root:x:0:0:0:/root:/bin/tcsh)

Add **-DSHADOWPW** to the etc/afpd/Makefile like so:

```
CFLAGS= ${DEFS} ${AFSDEFS} ${KRBDEFS} ${DESDEFS} ${OPTOPTS} ${INCPATH} \
        ${PAMDEFS} -DAPPLCNAME -DCRLF -DSHADOWPW # -DDOWNCASE
```

If you are running libc.so.5, comment out -lrpcsvc in sys/linux/Makefile. If you're using PAM, add -DUSE_PAM to this Makefile.

Install a DES library if you don't already have one and are planning on using encrypted passwords. (If not, comment the DES line out of the Makefile and skip this step.)

Get a copy of libdes from [\[ftp://ftp.psy.uq.oz.au/pub/Crypto/DES/libdes.tar.gz\]](ftp://ftp.psy.uq.oz.au/pub/Crypto/DES/libdes.tar.gz).

Untar it somewhere and do a make.

To optionally install libdes on your system (which isn't a half bad idea) type `make install`.

You will need the file `des.h` which should be there if the make was successful.

Copy `des.h` to your netatalk include directory: `cp des.h /path/netatalk/include/`.

Install TCP Wrappers if it's not already installed.

To compile, netatalk needs `tcpd.h` so if you don't already have it, download the TCP Wrappers package from [\[ftp://ftp.cert.org/pub/tools/tcp_wrappers/\]](ftp://ftp.cert.org/pub/tools/tcp_wrappers/).

Type `cp tcpd.h /path/netatalk/include/` to copy `tcpd.h` to your netatalk source include directory.

You probably have `tcpd` on your system. In the rare case that you don't, you'll want to compile it now by typing `make REAL_DAEMON_DIR=/usr/sbin linux` and install it. Look at the included README file for installation instructions.

Compile netatalk

The all important step: in your netatalk source directory, type `make`.

Make sure everything compiles up without an error. You may see some warnings which are safe to ignore, but if you see the compiler exit with some sort of error, try and see if you can figure out why it is dying. Remember, the compiler is going to look in your include directory for `tcpd.h` and `des.h` so make sure they are there.

If you are really having trouble with the compile, try the [tips](#) page. If you are **really** having trouble, you could download a pre-compiled version of netatalk from the download page.

[\[http://thehamptons.com/anders/netatalk/download.html\]](http://thehamptons.com/anders/netatalk/download.html)

Install netatalk

Type `make install` in the source directory and netatalk will be installed where you stated in the Makefile. (default is `/usr/local/ataalk`)

Edit /etc/services

Add the following lines to your `/etc/services` file:

```
rtmp      1/ddp      # Routing Table Maintenance Protocol
nbp       2/ddp      # Name Binding Protocol
echo      4/ddp      # AppleTalk Echo Protocol
zip       6/ddp      # Zone Information Protocol

afpovertcp 548/tcp    # AFP over TCP
afpovertcp 548/udp
```

Copy atalkd.conf

Type `cp conf/atalkd.conf /usr/local/ataalk/etc/`. The simplest config file for `atalkd` (like the one you just copied) is one with no commands in it.

`atalkd` is the "classic Appletalk" kernel interface. It will serve as the link between the kernel appletalk module and the rest of the classic Appletalk functions of netatalk. It will even take care of Appletalk routing between multiple network cards.

Copy afpd.conf file

Type `cp config/afpd.conf /usr/local/ataalk/etc`. This file sets Classic Appletalk and Appleshare IP server options. Leaving it alone is the easiest thing to do, however the default does not enable AppleshareIP and there are quite a few configurable options in this file so there is a section specific to it at:

[\[http://thehamptons.com/anders/netatalk/afpd.conf.html\]](http://thehamptons.com/anders/netatalk/afpd.conf.html)

Copy and edit AppleVolumes.*

Type `cp conf/AppleVolumes.default /usr/local/ataalk/etc` and `cp conf/AppleVolumes.system /usr/local/ataalk/etc`. The `AppleVolumes.default` and `AppleVolumes.system` files are listings of volume to path mappings and type / creator mappings. `AppleVolumes.default` will be parsed when a real user logs in. (ie: not the guest account) The settings in `AppleVolumes.default` can be overridden with an `AppleVolumes` or `.AppleVolumes` file in the users' home directory. `AppleVolumes.system` will be processed for all users. Adding the lines:

```
#
# volumes
~      Home
/music "Some Tunes"
#
# type / creator mappings
.txt TEXT MSWD
```

will cause two volumes to show up, one called **Home** which is the user's home directory and one called **Some Tunes** that is the directory `/music`. The rest of the file is a listing of "dot extensions" to type / creator mappings. A Mac file has two parts, a data fork and a resource fork. Unix files are only one long list of bits. Therefore, if you have the file `test.txt` created in Unix, looking at it on the Mac will reveal an "Unknown Document" icon unless there is an entry in the type / creator section of the `AppleVolumes` files. The last line in the above example will make a file ending in `.txt` show up as a Microsoft Word (MSWD) text (TEXT) document and will launch Word when double-clicked. (You actually register programs with Apple to have an official creator mapping assigned.) The default list of type / creator mappings in the `AppleVolumes.system` file is probably adequate unless you are planning on making many files in Unix and using them on Macs. One possible reason to use a large type / creator map is if you plan to export the same directory to Windows clients via Samba and you want Mac users to be able to double click PC files to open them.

Compile Appletalk in your kernel

There are two kinds of Appletalk servers you can run, and both can run at the same time. Classic Appletalk needs to run `atalkd` and `afpd` while Appletalk over TCP/IP only requires that `afpd` be running. `Atalkd` requires that Appletalk and TCP/IP support be compiled into your kernel while Appletalk IP only requires TCP/IP support. Many mainstream Linux distributions contain a kernel with Appletalk already compiled in and everyone probably has TCP/IP. You can check to see if your kernel supports Appletalk by typing:

```
dmesg | grep Apple
```

If you see anything pop up such as `Appletalk 0.17 for Linux NET3.035` you have Classic Appletalk compiled into your kernel. If not you could check to see if you have Appletalk inserted into the kernel via a module by typing: `lsmod` and seeing if Appletalk shows up there. If neither of these statements show Appletalk, then you have to compile Appletalk into the kernel and reboot, or compile Appletalk as a module and `insmod appletalk.o`. If you need help compiling Appletalk into the kernel, take a look at the Kernel-HOWTO. (usually included in major distributions in the docs directory `/usr/doc`)

Start the server

Now comes the moment of truth. Try to test a startup of the `netatalk` server by running one of the `rc.atalk` scripts in the root of the `netatalk` source tree. Type `./rc.atalk.sysv start` and give it a few seconds. This will start up a Classic Appletalk fileserver and Appletalk IP if you have any TCP options set in the `afpd.conf` file. `Atalkd` will take the longest to start as it checks out the network before registering itself.

Alternatively you could just run an Appletalk over TCP/IP server by having a TCP server configured in the `afpd.conf` file and launching `afpd` by hand:

```
/usr/local/atalk/etc/afpd -F /usr/local/atalk/etc/afpd.conf
```

If everything seems to have started up without complaining, go to a Mac and open the chooser. (under the Apple menu on the left hand side) Click on AppleShare and see if your `netatalk` server shows up. (for AppleShareIP you have to click the AppleShareIP button and type your machine IP or name in.

With some luck you will be presented with a password prompt. You can only log into the server with a non root account that has a valid shell and a password of 8 or less characters. Guest access is also permitted unless denied in `afpd.conf`. If you can't log in as anyone but guest you probably don't have shadow password support compiled into your `netatalk` executables. If you are able to login, you will get a list of volumes which you can select and mount.

Use one of the startup scripts included with `netatalk` to launch your server when your system boots. There are a variety of ways that a machine will boot up. Some have an `rc.local` file where you can specify the path to an `rc.atalk` script while others use a symbolic link with a naming scheme to determine the order in which startup scripts will run. Take a look at `/etc/rc.d/` or `/etc/init.d/` for startup scripts.

If you want to serve more than 5 connections, you must supply a max connections variable to `afpd`. The `-c` flag will set this.

```
afpd -c 25
```

Jonathan Benson passed along a SysV style startup script which takes configuration information from a config file which supports starting things in the background. Anyone who wants to use it should untar it when logged in as root in their root (ie: /) directory then edit the file `/etc/atalk/config`. You can download it below:

<http://thehamptons.com/anders/netatalk/mirror/atalk-sysv.tar.gz>

Other options are available in `atalkd` and `afpd` as well as a suite of tools for diagnosing and administrating a `netatalk` setup. Try `man atalkd` and `man afdp` for configuration options and look at the Utilities page at:

<http://thehamptons.com/anders/netatalk/utlis.html>

Enjoy!

Paul Hargrove wrote an interesting package called `hfs` which will let you mount and have read / write access to Macintosh volumes. (Hard drives, floppys and CD-ROMs) If you install his filesystem module and mount a volume with the `fork=netatalk` option, you will be able to access it through `netatalk`. His module deals with resource information properly so icons show up correctly. <http://www.sccm.stanford.edu/Students/hargrove/HFS/>

Original document: <http://thehamptons.com/anders/netatalk/>